

EDA FINAL PROJECT

Name: Bukkarayasamudram Naga Padma Maanasa

Reg.no: 12016660

Section: K20KT

Sub: INT-353

Project name: Video Game Sales

Git Hub Link:

<https://github.com/B-Maanasa/Video-Game-Sales>

Kaggle Data set Link:

[https://www.kaggle.com/datasets/gregorut/videogamesale](https://www.kaggle.com/datasets/gregorut/videogamesales)

[s](#)

The reason why I choose the "Video Game Sales" dataset:

Video games are a billion-dollar business and have been for many years. In 2020, the revenue from the worldwide PC gaming market was estimated at almost 37 billion U.S. dollars, while the mobile gaming market generated an estimated income of over 77 billion U.S. dollars. What is significant nowadays

is that the first generation of gamers is now grown up and has considerable spending power at its disposal. Despite the high average daily time spent playing games among kids, the hobby can no longer be solely child's play. It

was found that video gaming is gaining popularity among parents worldwide as well, with a relatively even split in terms of the gender distribution of

video gaming parents worldwide.

Information on Video Game Sales data set:

This data set contains the list of video games with sales greater than 100,000

copies from the year 1980 to 2016. There are 16,598 records(rows). 2 forms were

dropped due to incomplete information.

There are 11 fields (columns) included:

This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com.

Fields include-

Rank - Ranking of overall sales

Name - The games name

Platform - Platform of the game's release (i.e. PC, PS4, etc.)

Year - Year of the game's release

Genre - Genre of the game

Publisher - Publisher of the game

NA_Sales - Sales in North America (in millions)

EU_Sales - Sales in Europe (in millions)

JP_Sales - Sales in Japan (in millions)

Other_Sales - Sales in the rest of the world (in millions)

Global_Sales - Total worldwide sales.

What I do in my analysis:

- First of all, I'll clean my data set accordingly to understand it better.

Top platforms with the highest number of gaming sales

- Platforms sales by year.
- Famous platforms sales in each continent separately.
- Global sales

In Publishers

- Publisher sales by year.
- Famous publisher sales in each continent separately.
- global sales number per year

Sales of genres over the years to find the popular genres.

- Famous genres sales in each continent separately.
- Global sales.
- genres published by each publisher.

In which year there are more sales?

Starting with the project:

First I will start with importing the libraries:

Importing the Libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing and Displaying the data set:

Importing the data set

```
In [2]: Gaming = pd.read_csv("Gaming.csv")
```

Displaying the dataset

```
In [3]: Gaming
```

```
Out[3]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37
...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00	0.00	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01

16598 rows × 11 columns

Listing out the first 10 and last 5 rows:

Listing out first 10 rows

```
In [4]: Gaming.head(10)
```

Out[4]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37
5	6	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26
6	7	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	2.90	30.01
7	8	Wii Play	Wii	2006.0	Misc	Nintendo	14.03	9.20	2.93	2.85	29.02
8	9	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.59	7.06	4.70	2.26	28.62
9	10	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31

Listing out last 5 rows

```
In [5]: Gaming.tail()
```

Out[5]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.0	0.0	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.0	0.0	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.0	0.0	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.0	0.0	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.0	0.0	0.01

Information Regarding the data set:

Dimensions of the data set:

```
In [6]: numberofrows_1, numberofcolumns_1 = Gaming.shape  
print('There are {} "rows and {} "columns'.format(numberofrows_1, numberofcolumns_1)+' in my Animes Dataset.')
```

There are "16598 "rows and 11 "columns in my Animes Dataset.

Describing the data set:

```
In [7]: Gaming.describe()
```

```
Out[7]:
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0.537441
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.555028
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.470000
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.740000

About the columns:

Displaying the columns:

```
In [8]: Gaming.columns
```

```
Out[8]: Index(['Rank', 'Name', 'Platform', 'Year', 'Genre', 'Publisher', 'NA_Sales',  
              'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales'],  
             dtype='object')
```

Data regarding columns:

```
In [9]: Gaming.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 16598 entries, 0 to 16597  
Data columns (total 11 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   Rank            16598 non-null  int64  
1   Name            16598 non-null  object  
2   Platform        16598 non-null  object  
3   Year            16327 non-null  float64  
4   Genre           16598 non-null  object  
5   Publisher       16540 non-null  object  
6   NA_Sales        16598 non-null  float64  
7   EU_Sales        16598 non-null  float64  
8   JP_Sales        16598 non-null  float64  
9   Other_Sales     16598 non-null  float64  
10  Global_Sales    16598 non-null  float64  
dtypes: float64(6), int64(1), object(4)  
memory usage: 1.4+ MB
```

Finding the null and duplicate values:

Finding the number of null values in each columns:

```
In [10]: Gaming.isnull().sum()
```

```
Out[10]: Rank          0  
Name          0  
Platform      0  
Year         271  
Genre         0  
Publisher     58  
NA_Sales      0  
EU_Sales      0  
JP_Sales      0  
Other_Sales   0  
Global_Sales  0  
dtype: int64
```

As the null values are less I'm leaving it the same

Finding the duplicate values:

```
In [11]: Gaming.duplicated().sum()
```

```
Out[11]: 0
```

There are no duplicate values in the dataset

Displaying the correlation heatmap:

```
Out[72]: <function matplotlib.pyplot.show(close=None, block=None)>
```



NA_Sales and EU_sales have much effect in Global_Sales. And NA_Sales and EU_sales are mostly correlated

Displaying the no of unique games available in the data set:

```
Out[13]: Need for Speed: Most Wanted      12
          Ratatouille                      9
          FIFA 14                          9
          LEGO Marvel Super Heroes         9
          Madden NFL 07                   9
          ..
          Ar tonelico Qoga: Knell of Ar Ciel 1
          Galaga: Destination Earth         1
          Nintendo Presents: Crossword Collection 1
          TrackMania: Build to Race         1
          Know How 2                        1
          Name: Name, Length: 11493, dtype: int64
```

- According to the above info, we have 11493 unique games available in the data set.

The data set contains the years from 1980 to 2020:

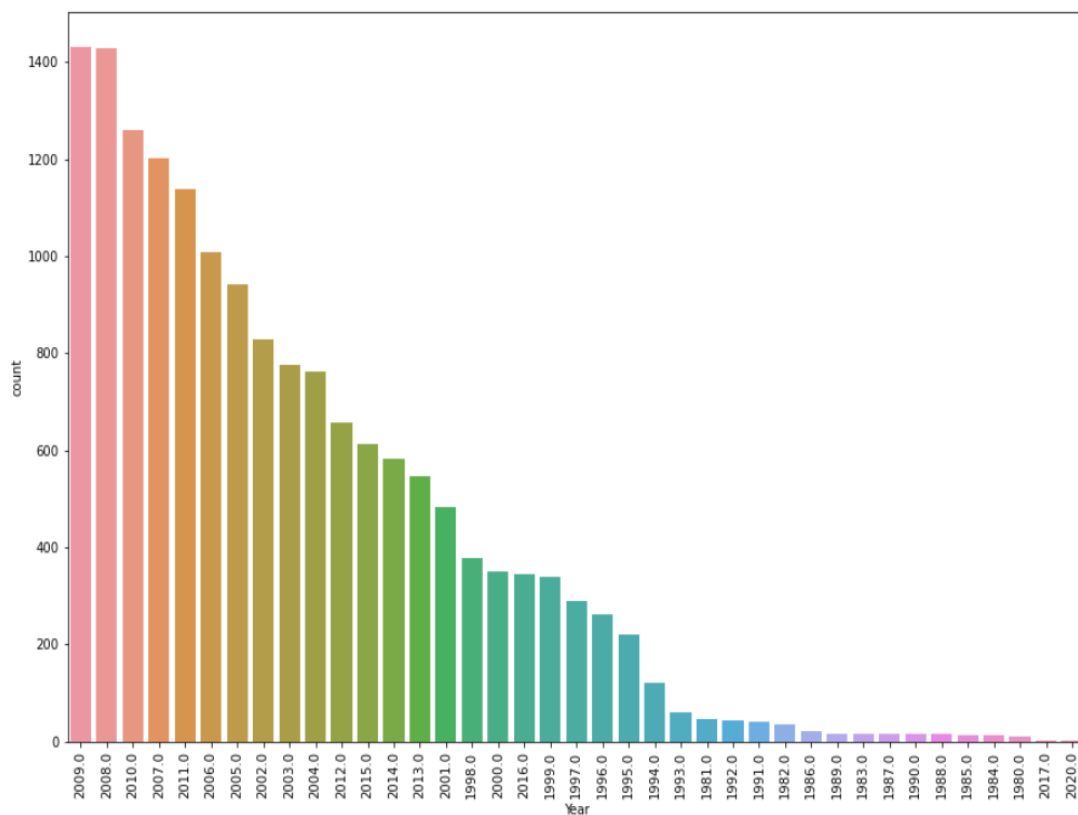
```
Out[15]: 2009.0      1431
          2008.0      1428
          2010.0      1259
          2007.0      1202
          2011.0      1139
          2006.0      1008
          2005.0       941
          2002.0       829
          2003.0       775
          2004.0       763
          2012.0       657
          2015.0       614
          2014.0       582
          2013.0       546
          2001.0       482
          1998.0       379
          2000.0       349
          2016.0       344
          1999.0       338
          1997.0       289
          1996.0       263
          1995.0       219
          1994.0       121
          1993.0        60
          1981.0        46
          1992.0        43
          1991.0        41
          1982.0        36
          1986.0        21
          1989.0        17
          1983.0        17
          1990.0        16
          1987.0        16
          1988.0        15
          1985.0        14
          1984.0        14
          1980.0         9
          2017.0         3
          2020.0         1
          Name: Year, dtype: int64
```


Unique and Null values of the year:

```
In [17]: print('No. of unique values in Year: ', Gaming['Year'].nunique())
          print('N/a values: ', Gaming['Year'].isna().sum())
```

No. of unique values in Year: 39
N/a values: 271

Plotting Counter plot to know in which year we have the most no of gaming sales:

[illegible]

Top platform:

```
Out[20]: DS      2163
         PS2     2161
         PS3     1329
         Wii     1325
         X360    1265
         PSP     1213
         PS      1196
         PC      960
         XB      824
         GBA     822
         GC      556
         3DS     509
         PSV     413
         PS4     336
         N64     319
         SNES    239
         XOne    213
         SAT     173
         WiiU    143
         2600    133
         NES     98
         GB      98
         DC      52
         GEN     27
         NG      12
         SCD     6
         WS      6
         3DO     3
         TG16    2
         GG      1
         PCFX    1
         Name: Platform, dtype: int64
```

- DS and PS2 are the top two platforms

There are 31 unique values and 0 null values:

Checking Unique values:

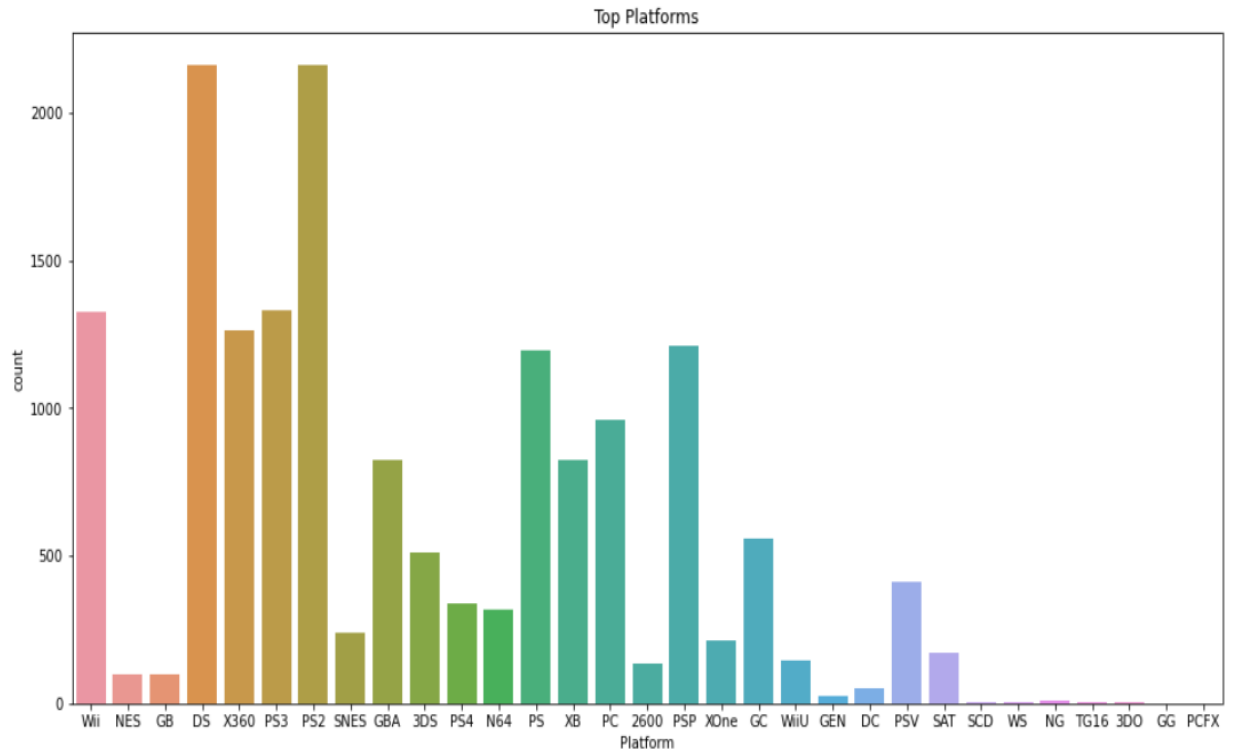
```
In [21]: print('No. of unique values in platform: ', Gaming['Platform'].nunique())
         print('N/a values: ', Gaming['Platform'].isna().sum())
```

```
No. of unique values in platform: 31
N/a values: 0
```

Showing the Platform using Countplot

```
In [22]: plt.figure(figsize=(16,8))
sns.countplot(data=Gaming,x='Platform',)
plt.title('Top Platforms')
```

```
Out[22]: Text(0.5, 1.0, 'Top Platforms')
```



Finding the number of unique values in Genre:

GENRE

```
In [23]: Gaming.Genre.unique()
```

```
Out[23]: array(['Sports', 'Platform', 'Racing', 'Role-Playing', 'Puzzle', 'Misc',
               'Shooter', 'Simulation', 'Action', 'Fighting', 'Adventure',
               'Strategy'], dtype=object)
```

Checking Unique values:

```
In [24]: print('No. of unique values in Genre: ', Gaming['Genre'].nunique())
print('N/a values: ', Gaming['Genre'].isna().sum())
```

```
No. of unique values in Genre: 12
N/a values: 0
```

Finding which genre game has been popular

```
In [25]: Gaming['Genre'].value_counts()
```

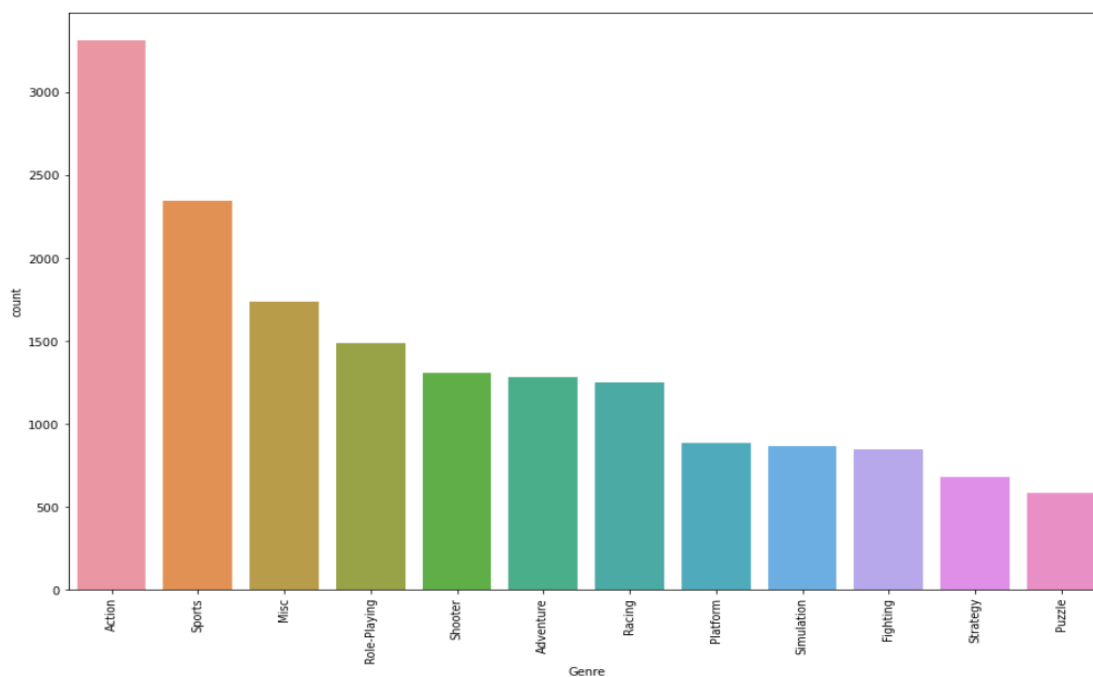
```
Out[25]: Action      3316  
Sports      2346  
Misc        1739  
Role-Playing 1488  
Shooter     1310  
Adventure   1286  
Racing      1249  
Platform     886  
Simulation   867  
Fighting     848  
Strategy     681  
Puzzle       582  
Name: Genre, dtype: int64
```

Action and Sports are in first and second place

Showing the top Genre using Countplot

```
In [26]: plt.figure(figsize=(15, 10))  
sns.countplot(x="Genre", data=Gaming, order = Gaming['Genre'].value_counts().index)  
plt.xticks(rotation=90)
```

```
Out[26]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),  
[Text(0, 0, 'Action'),  
Text(1, 0, 'Sports'),  
Text(2, 0, 'Misc'),  
Text(3, 0, 'Role-Playing'),  
Text(4, 0, 'Shooter'),  
Text(5, 0, 'Adventure'),  
Text(6, 0, 'Racing'),  
Text(7, 0, 'Platform'),  
Text(8, 0, 'Simulation'),  
Text(9, 0, 'Fighting'),  
Text(10, 0, 'Strategy'),  
Text(11, 0, 'Puzzle')])
```



Finding which publisher companies manufactured most of the games

```
In [27]: Gaming['Publisher'].value_counts()

Out[27]: Electronic Arts      1351
         Activision          975
         Namco Bandai Games   932
         Ubisoft             921
         Konami Digital Entertainment  832
         ...
         Warp                 1
         New                  1
         Elite                1
         Evolution Games      1
         UIG Entertainment    1
         Name: Publisher, Length: 578, dtype: int64
```

Electronic Arts, Activision are top two publisher companies.

Checking Unique values:

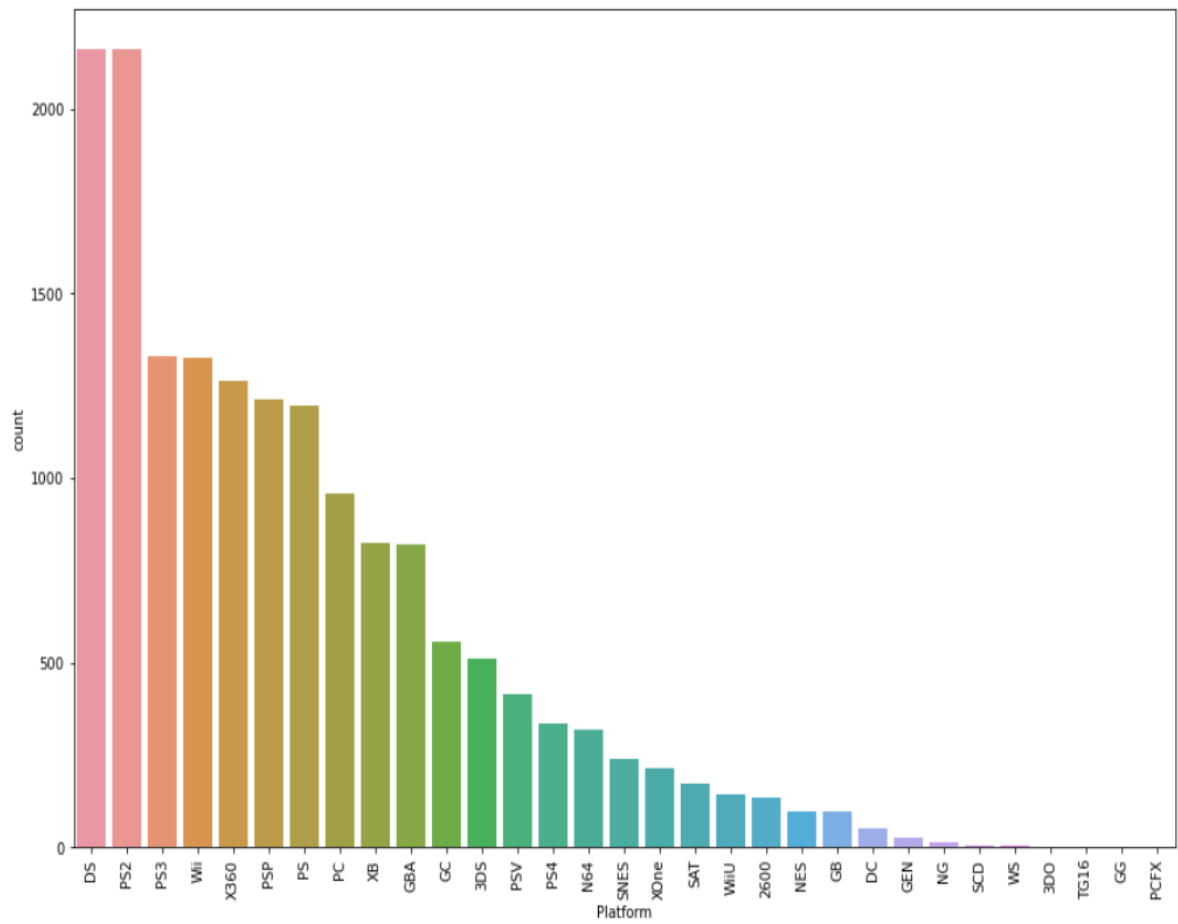
```
In [28]: print('No. of unique values in Publisher: ', Gaming['Publisher'].nunique())
         print('N/a values: ', Gaming['Publisher'].isna().sum())

No. of unique values in Publisher: 578
N/a values: 58
```

Showing the top Platform using Countplot

```
In [29]: plt.figure(figsize=(15, 10))
sns.countplot(x="Platform", data=Gaming, order = Gaming['Platform'].value_counts().index)
plt.xticks(rotation=90)
```

```
Out[29]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]),
 [Text(0, 0, 'DS'),
  Text(1, 0, 'PS2'),
  Text(2, 0, 'PS3'),
  Text(3, 0, 'Wii'),
  Text(4, 0, 'X360'),
  Text(5, 0, 'PSP'),
  Text(6, 0, 'PS'),
  Text(7, 0, 'PC'),
  Text(8, 0, 'XB'),
  Text(9, 0, 'GBA'),
  Text(10, 0, 'GC'),
  Text(11, 0, '3DS'),
  Text(12, 0, 'PSV'),
  Text(13, 0, 'PS4'),
  Text(14, 0, 'N64'),
  Text(15, 0, 'SNES'),
  Text(16, 0, 'XOne'),
  Text(17, 0, 'SAT'),
  Text(18, 0, 'WiiU'),
  Text(19, 0, '2600'),
  Text(20, 0, 'NES'),
  Text(21, 0, 'GB'),
  Text(22, 0, 'DC'),
  Text(23, 0, 'GEN'),
  Text(24, 0, 'NG'),
  Text(25, 0, 'SCD'),
  Text(26, 0, 'WS'),
  Text(27, 0, '3DO'),
  Text(28, 0, 'TG16'),
  Text(29, 0, 'GG'),
  Text(30, 0, 'PCFX')])
```



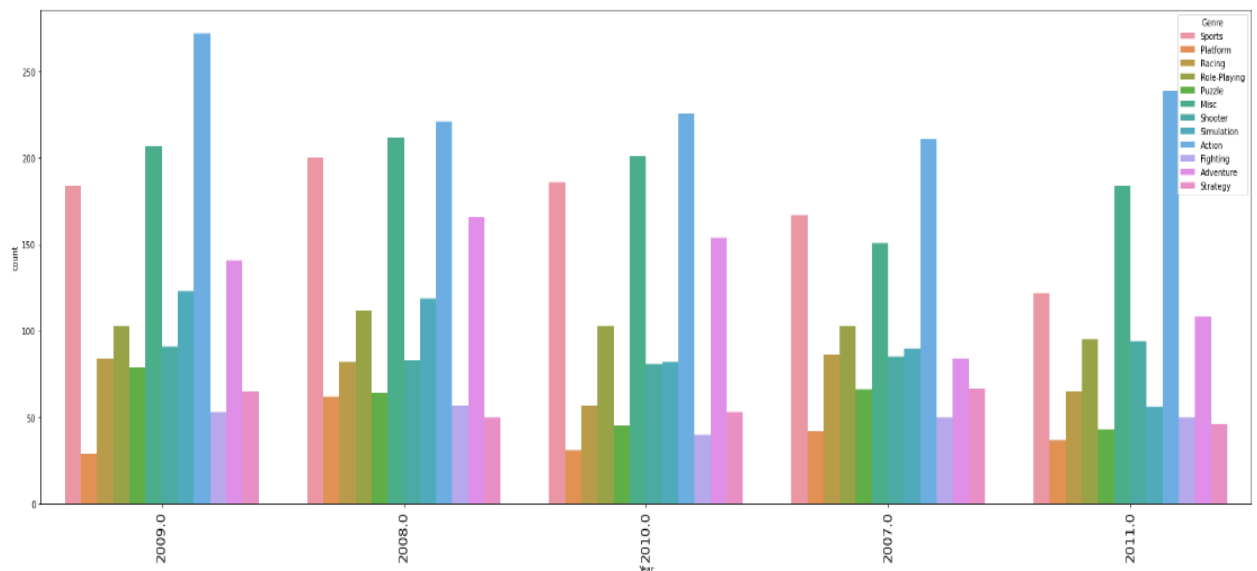
- DS and PS2 are the top platforms

Top 5-year Game Genre Releases:

Top 5 year Game Releases:

```
In [30]: plt.figure(figsize=(30, 10))
sns.countplot(x="Year", data=Gaming, hue="Genre", order=Gaming.Year.value_counts().iloc[:5].index)
plt.xticks(size=16, rotation=90)
```

```
Out[30]: (array([0, 1, 2, 3, 4]),
 [Text(0, 0, '2009.0'),
  Text(1, 0, '2008.0'),
  Text(2, 0, '2010.0'),
  Text(3, 0, '2007.0'),
  Text(4, 0, '2011.0')])
```



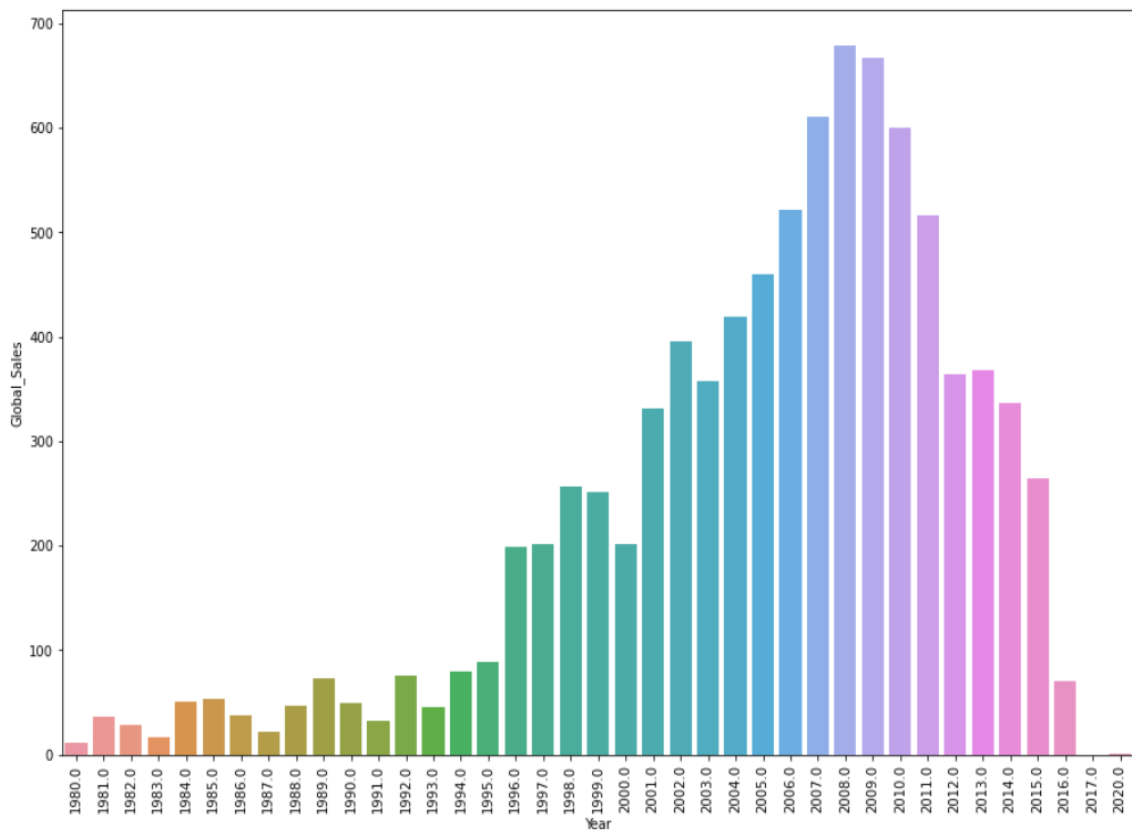
- 2009 Stand at the top

Finding which year has the highest sales globally

```
In [31]: Gaming_year = Gaming.groupby(by=['Year'])['Global_Sales'].sum()
Gaming_year = Gaming_year.reset_index()
```


Plotting the counterplot of the highest sales globally:

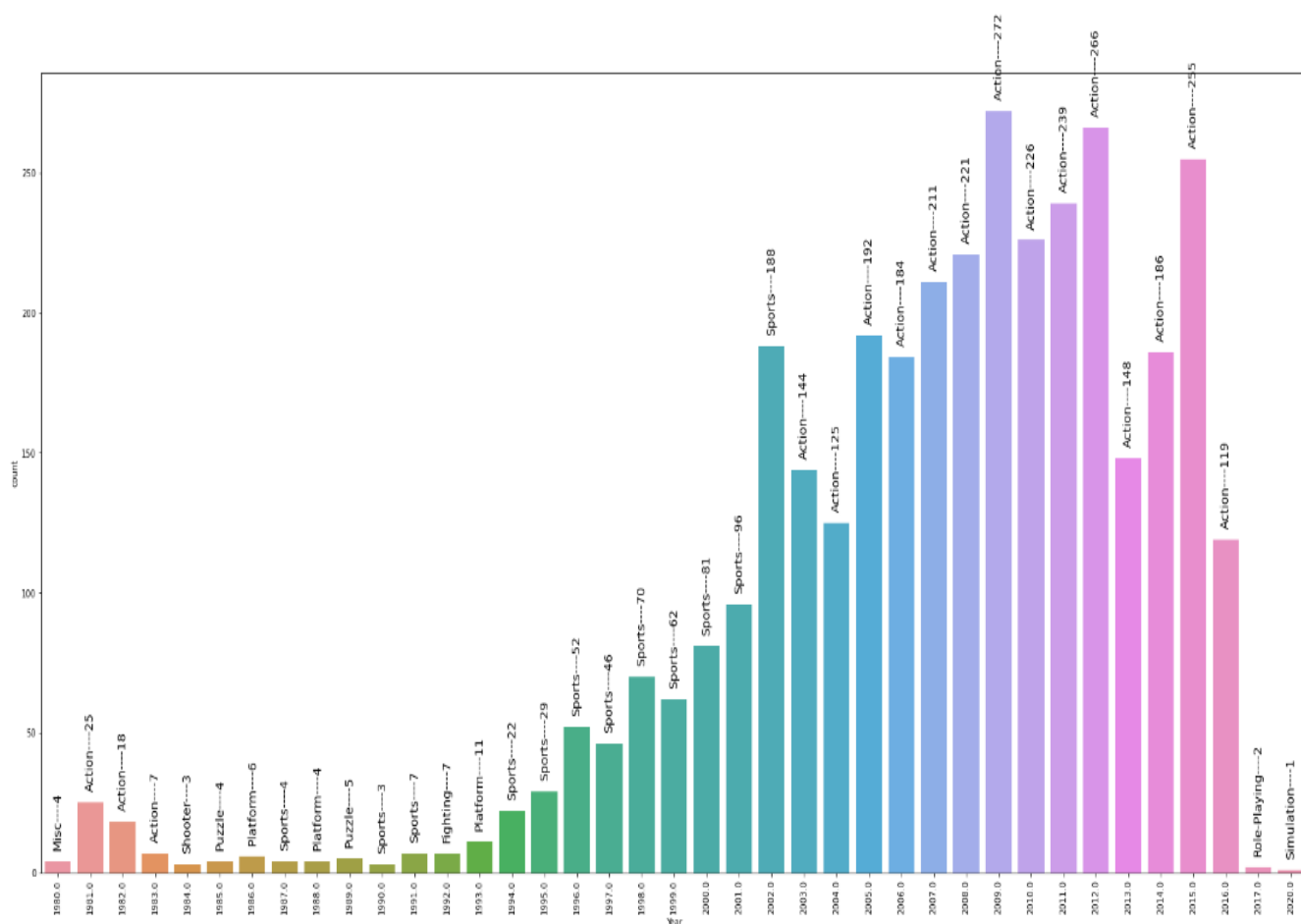
```
Out[32]: (array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12., 13., 14., 15., 16.,
 17., 18., 19., 20., 21., 22., 23., 24., 25., 26., 27., 28., 29., 30., 31., 32., 33.,
 34., 35., 36., 37., 38]),
 [Text(0, 0, '1980.0'),
  Text(1, 0, '1981.0'),
  Text(2, 0, '1982.0'),
  Text(3, 0, '1983.0'),
  Text(4, 0, '1984.0'),
  Text(5, 0, '1985.0'),
  Text(6, 0, '1986.0'),
  Text(7, 0, '1987.0'),
  Text(8, 0, '1988.0'),
  Text(9, 0, '1989.0'),
  Text(10, 0, '1990.0'),
  Text(11, 0, '1991.0'),
  Text(12, 0, '1992.0'),
  Text(13, 0, '1993.0'),
  Text(14, 0, '1994.0'),
  Text(15, 0, '1995.0'),
  Text(16, 0, '1996.0'),
  Text(17, 0, '1997.0'),
  Text(18, 0, '1998.0'),
  Text(19, 0, '1999.0'),
  Text(20, 0, '2000.0'),
  Text(21, 0, '2001.0'),
  Text(22, 0, '2002.0'),
  Text(23, 0, '2003.0'),
  Text(24, 0, '2004.0'),
  Text(25, 0, '2005.0'),
  Text(26, 0, '2006.0'),
  Text(27, 0, '2007.0'),
  Text(28, 0, '2008.0'),
  Text(29, 0, '2009.0'),
  Text(30, 0, '2010.0'),
  Text(31, 0, '2011.0'),
  Text(32, 0, '2012.0'),
  Text(33, 0, '2013.0'),
  Text(34, 0, '2014.0'),
  Text(35, 0, '2015.0'),
  Text(36, 0, '2016.0'),
  Text(37, 0, '2017.0'),
  Text(38, 0, '2020.0')])
```



- 2008.0 678.90

- 2009.0 667.30
- 2007.0 611.13
- 2010.0 600.45
- 2006.0 521.04
- 2008 is at the top and 2009 is second

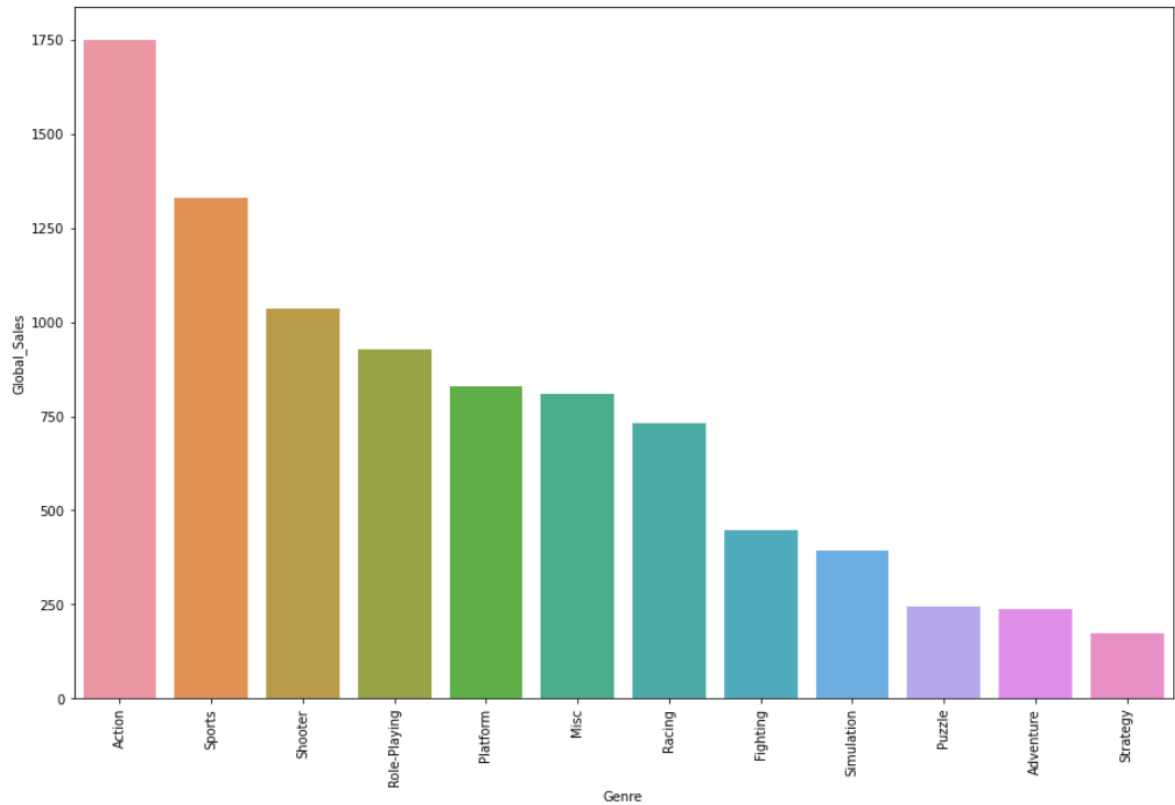
Comparing Year and Genre- we can find which genre is released most in each year:



As you can see, in 2009 Action Genre is released which is placed top with 272 game and in 2012 again Action, placed in 2nd with 266 games release.

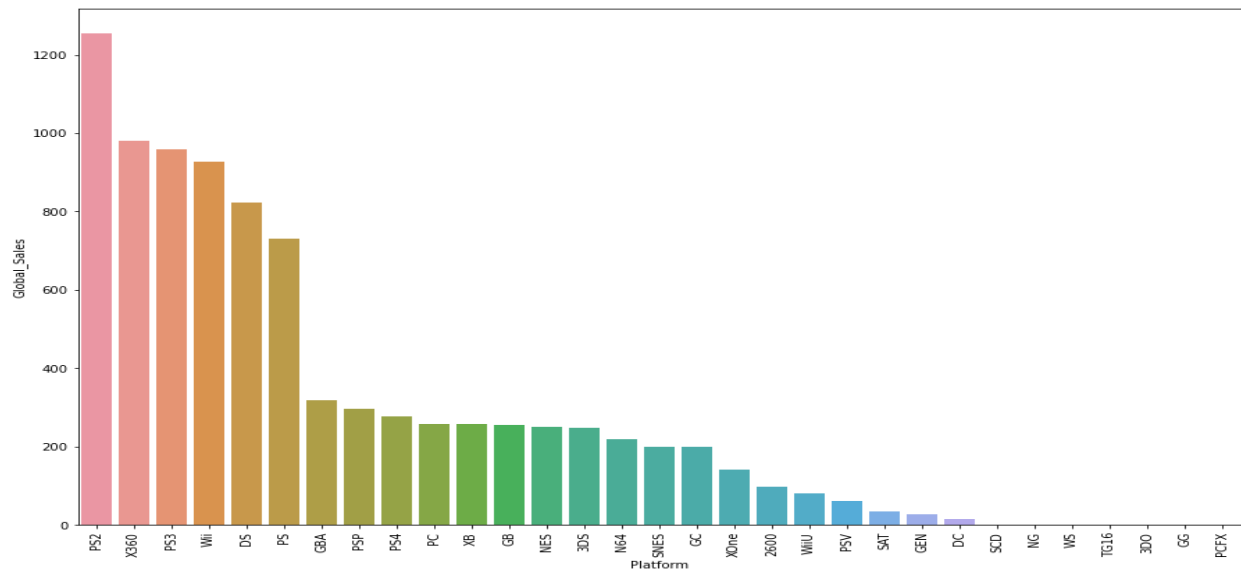
Finding which genre game has the highest sale price globally:

```
Out[37]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
          [Text(0, 0, 'Action'),
           Text(1, 0, 'Sports'),
           Text(2, 0, 'Shooter'),
           Text(3, 0, 'Role-Playing'),
           Text(4, 0, 'Platform'),
           Text(5, 0, 'Misc'),
           Text(6, 0, 'Racing'),
           Text(7, 0, 'Fighting'),
           Text(8, 0, 'Simulation'),
           Text(9, 0, 'Puzzle'),
           Text(10, 0, 'Adventure'),
           Text(11, 0, 'Strategy')])
```



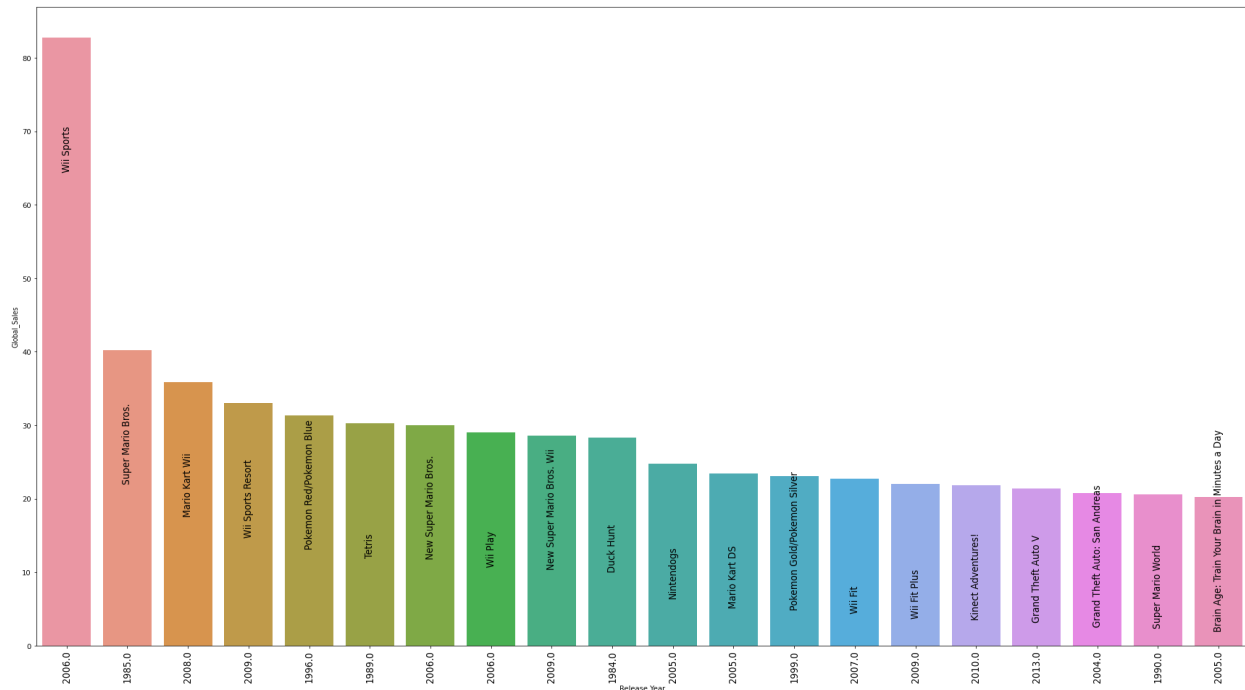
Looks like Action and sports are in top two globally

Finding the platform with the highest sale price globally:



- **First is PS2, and Second is X360.**

The specific game with the highest sales globally:



- As we can see, 2006 ranks the top with the game Wii sports.

The Publisher who has the highest sale price globally:

(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]),

[Text(0, 0, 'Nintendo'),

Text(1, 0, 'Electronic Arts'),

Text(2, 0, 'Activision'),

Text(3, 0, 'Sony Computer Entertainment'),

Text(4, 0, 'Ubisoft'),

Text(5, 0, 'Take-Two Interactive'),

Text(6, 0, 'THQ'),

Text(7, 0, 'Konami Digital Entertainment'),

Text(8, 0, 'Sega'),

Text(9, 0, 'Namco Bandai Games'),

Text(10, 0, 'Microsoft Game Studios'),

Text(11, 0, 'Capcom'),

Text(12, 0, 'Atari'),

Text(13, 0, 'Warner Bros. Interactive Entertainment'),

Text(14, 0, 'Square Enix'),

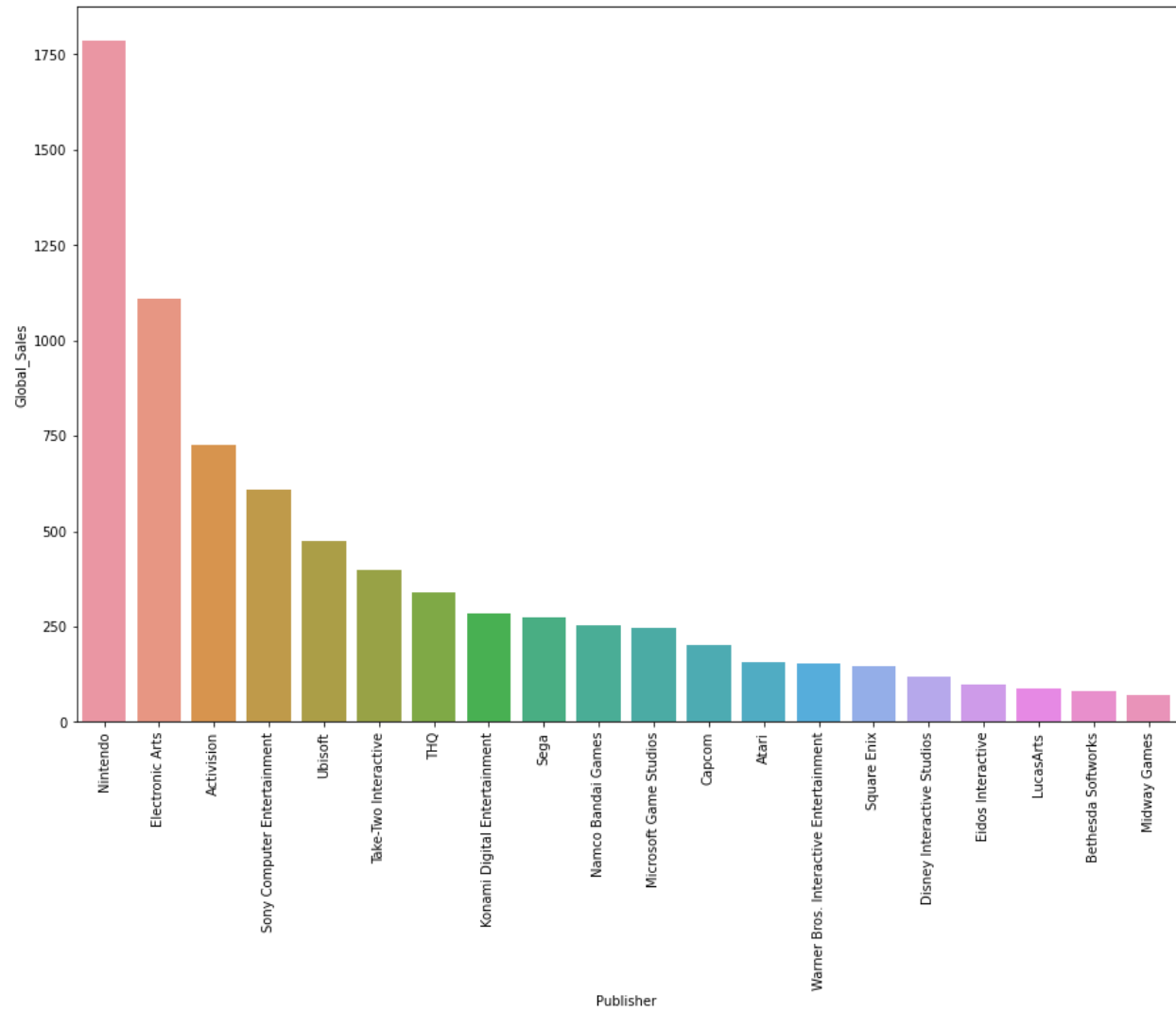
Text(15, 0, 'Disney Interactive Studios'),

Text(16, 0, 'Eidos Interactive'),

Text(17, 0, 'LucasArts'),

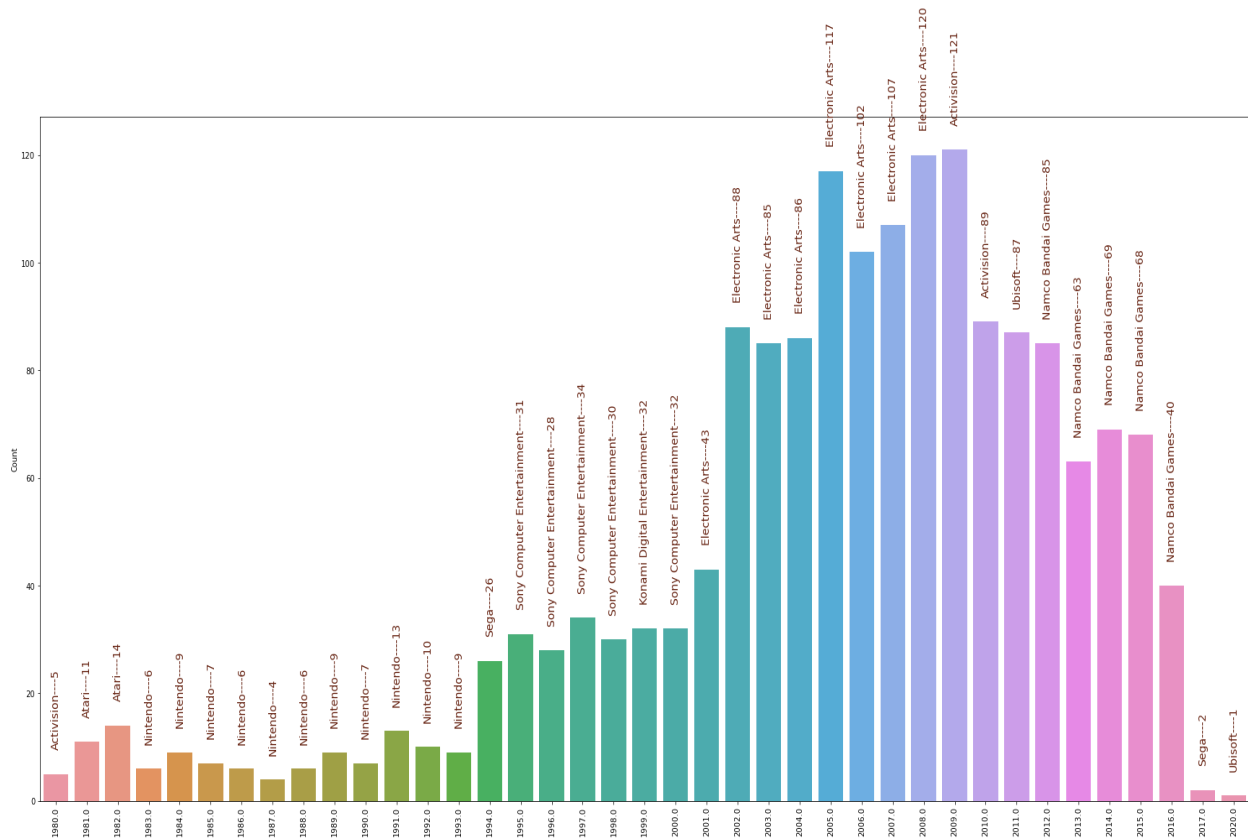
Text(18, 0, 'Bethesda Softworks'),

Text(19, 0, 'Midway Games'))]



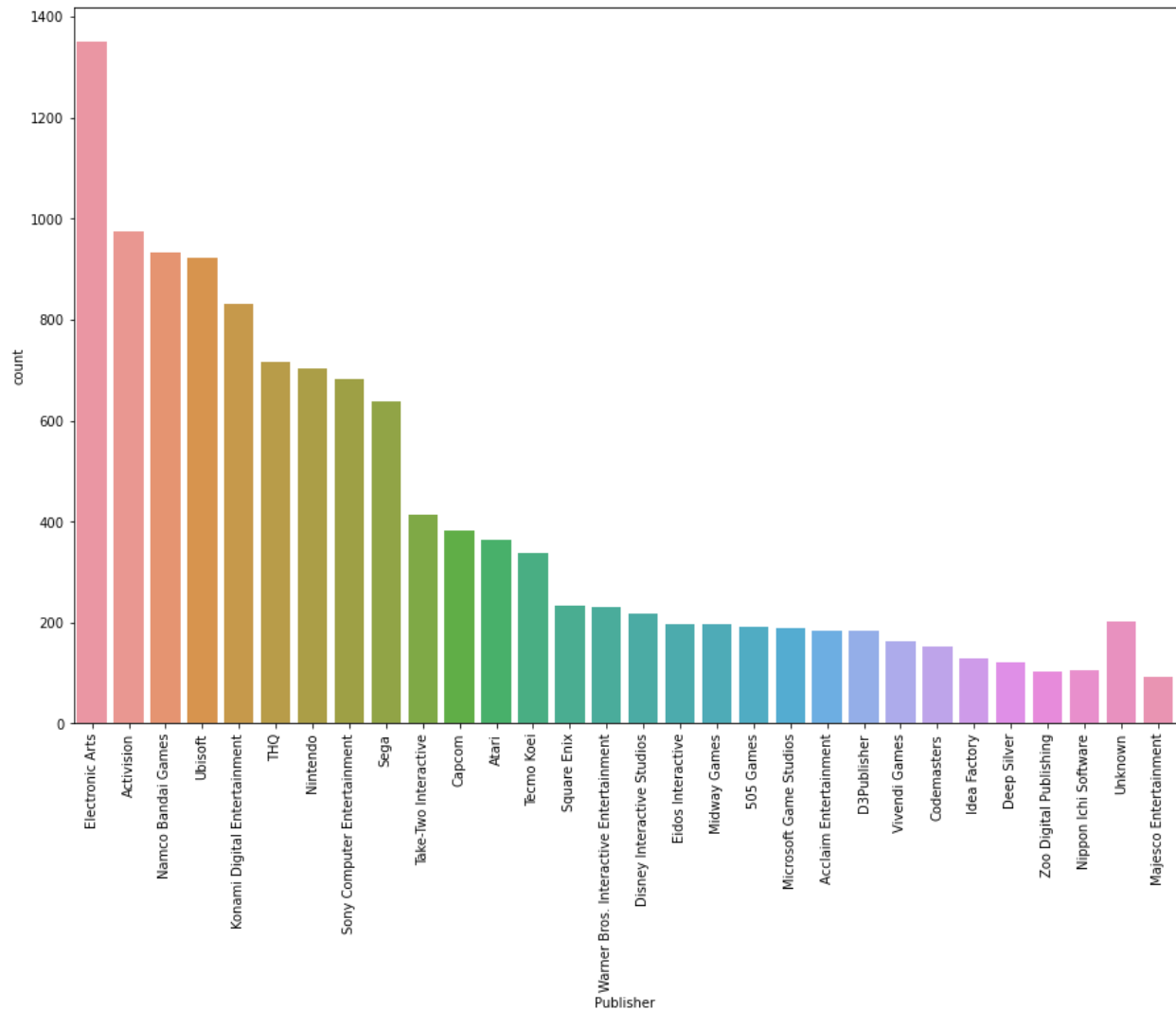
- **Nintendo and Electronic Arts are the top 2 publishers**

Comparing Year with Publisher, we can find which Publisher released the most games in each year:



- 2009 stands first with Activision-121, and the second is Electronic Arts -120 in 2008.

Now let's find out the top 30 publishers:

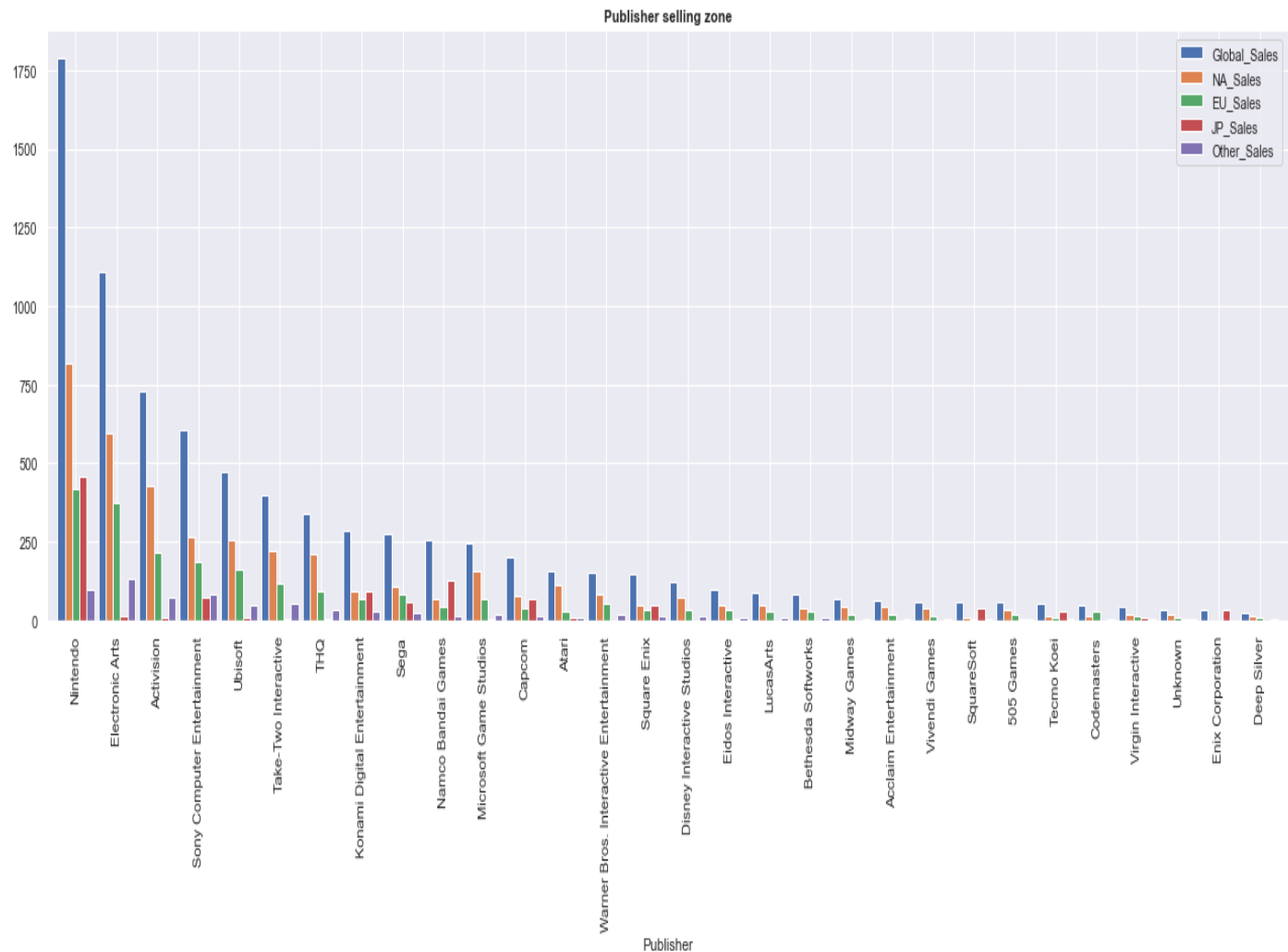


- Among the top 30 publishers, “Electronic Arts” stands at the top.

Conclusion:

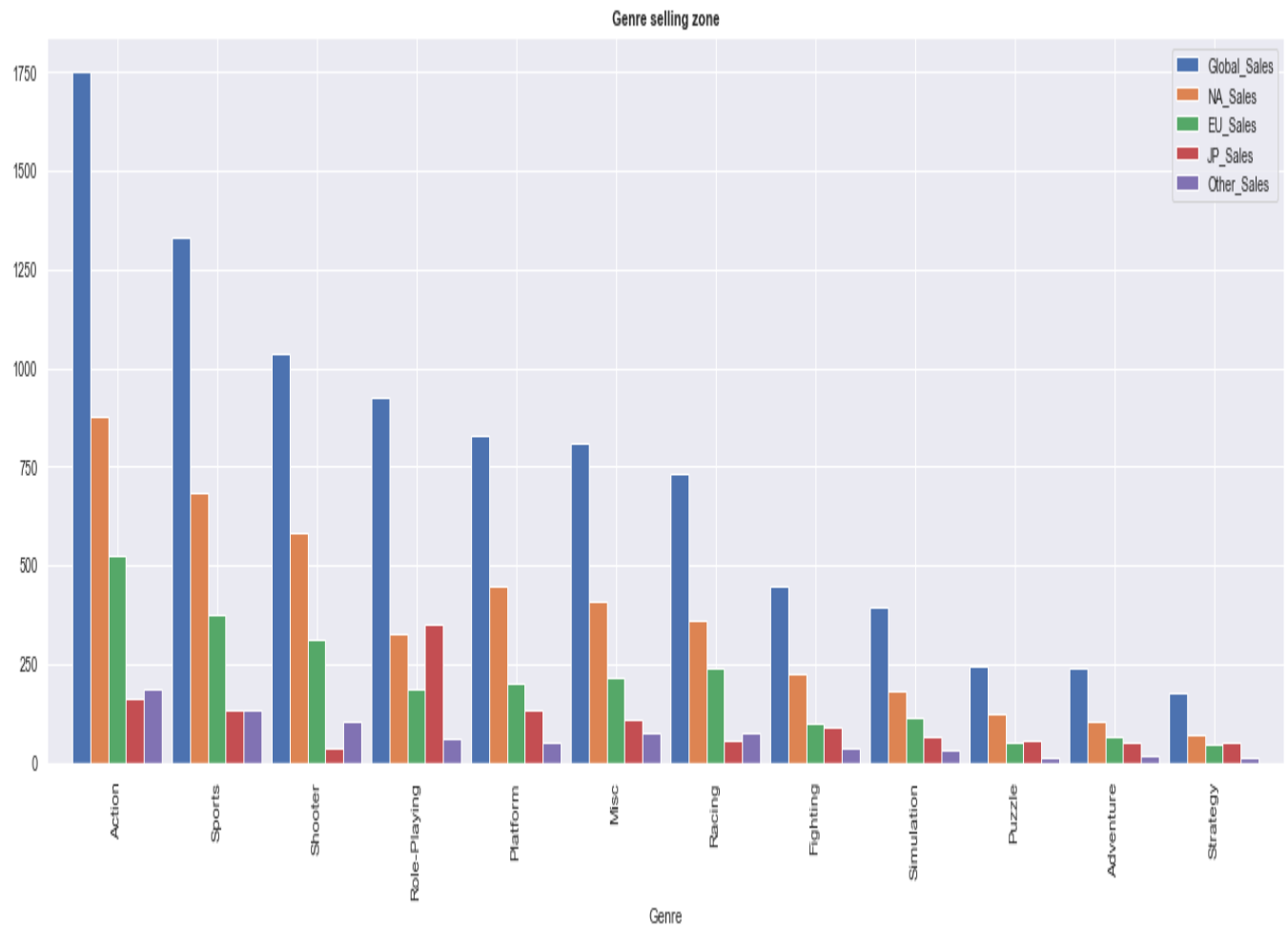
I'm concluding the report by showing the all sales comparison globally:

Global sales- Publisher comparison:



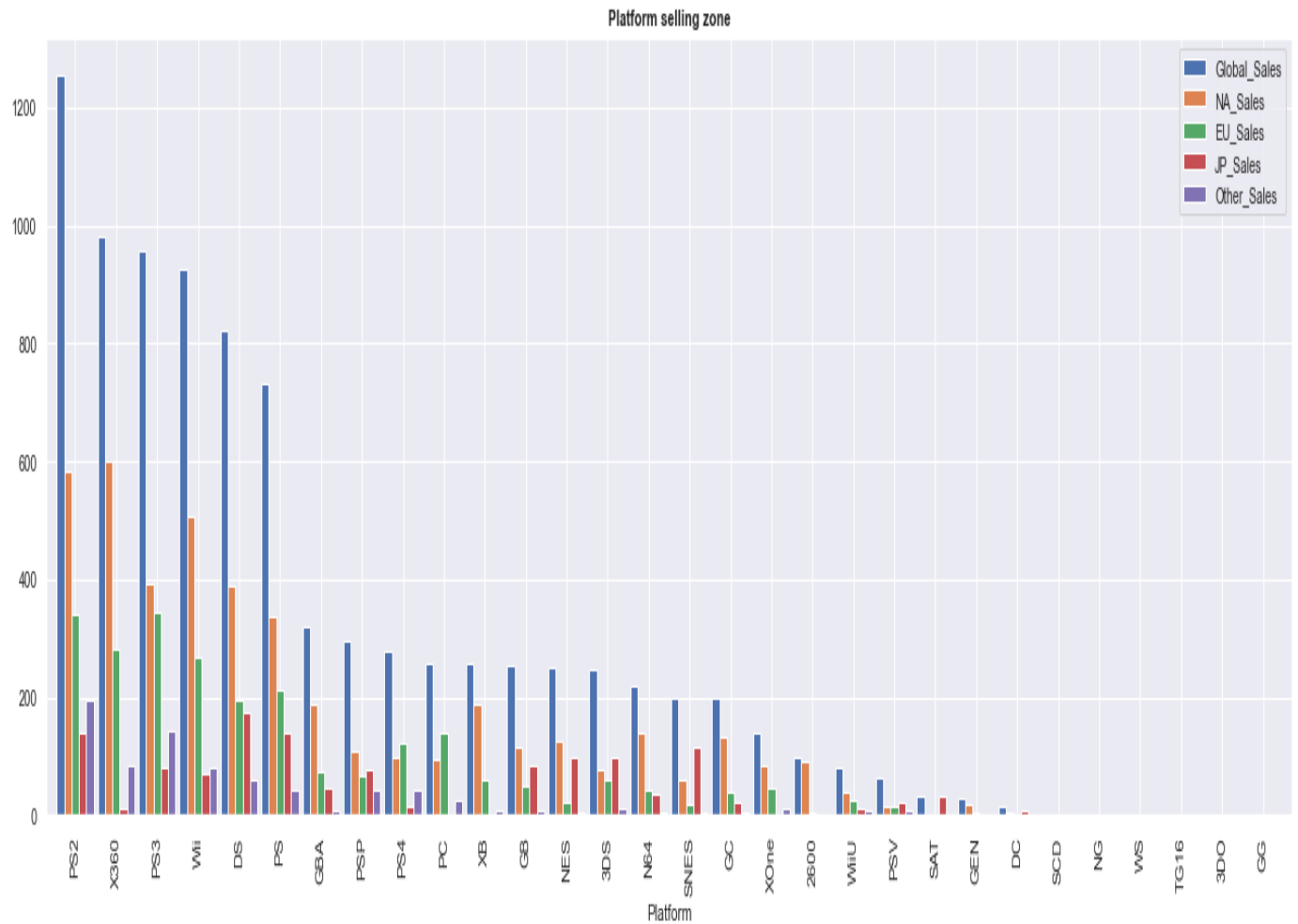
- Nintendo is at the top globally.

Global sales- Genre comparison:



- Action- globally stands at the top.

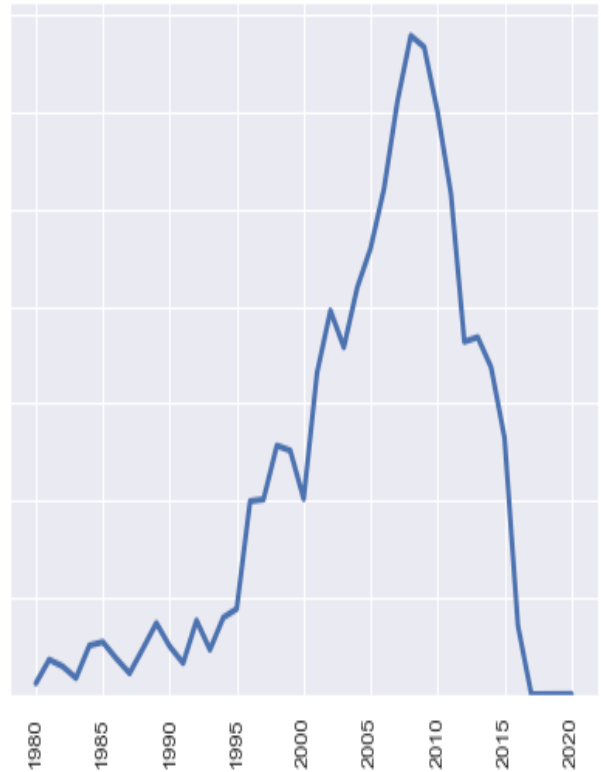
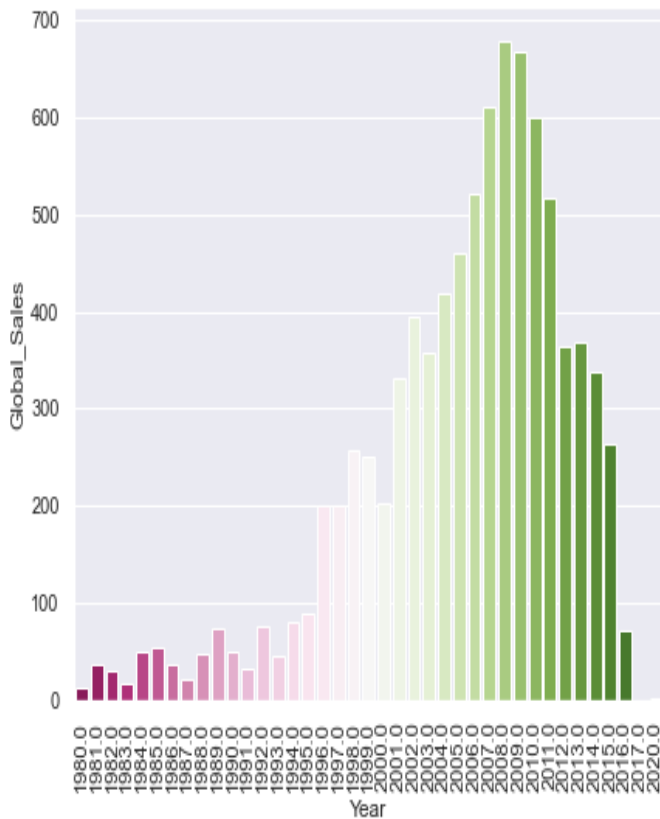
Global sales- platform comparison:



- PS2 is in first place globally.

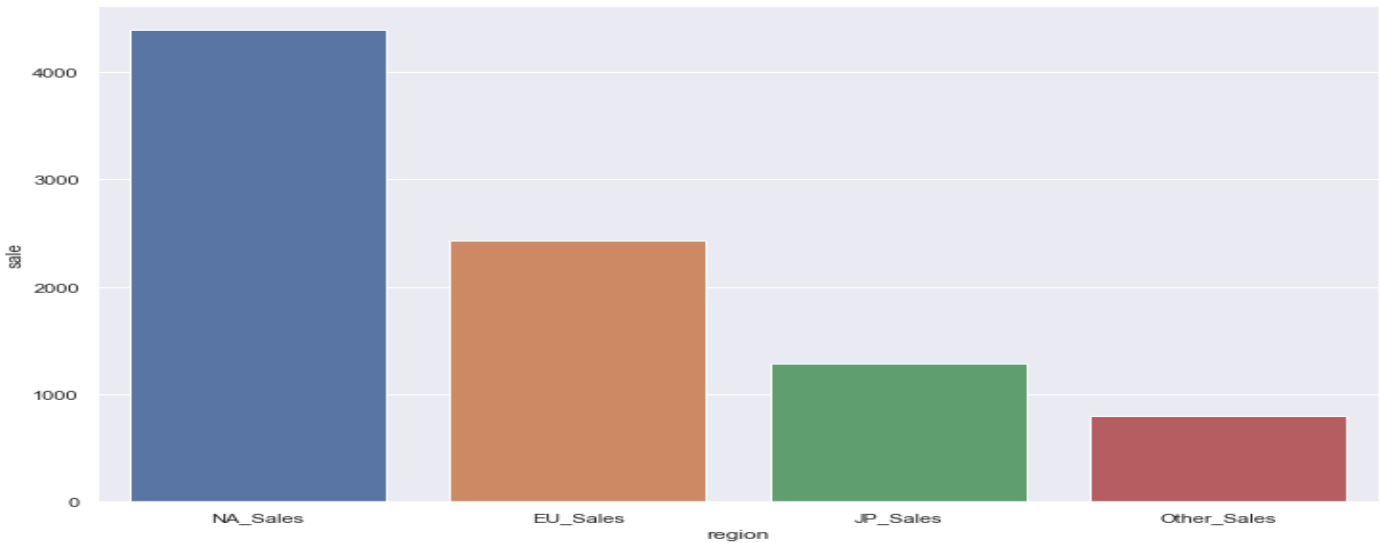
Global Sales of Games:

Global Sales Of Games (In Million)



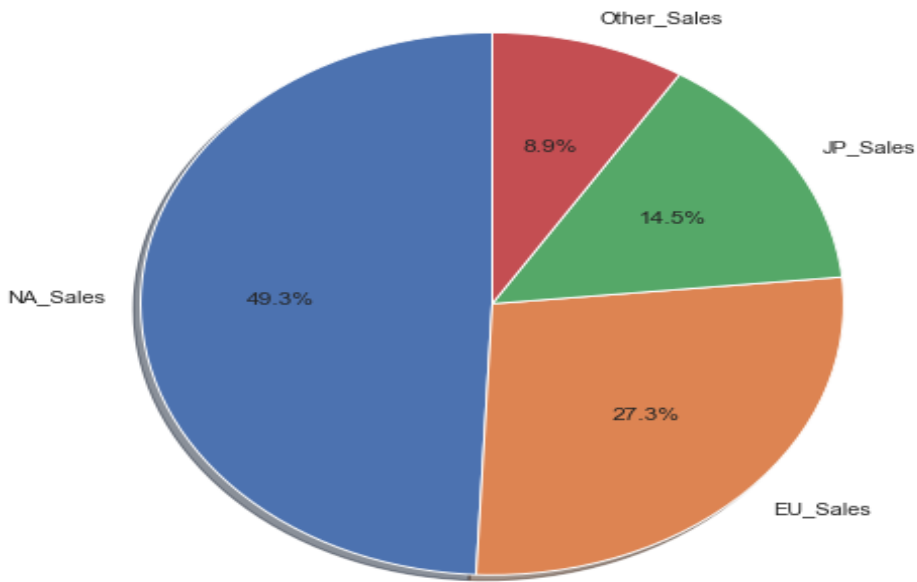
- As shown in the graph, 2008 had the highest sales globally and if we observe the line plot above, the sales of video games started growing from 2000 till the 2010s and gradually started decreasing after. and gradually decreasing after.

Total Revenue by Region:

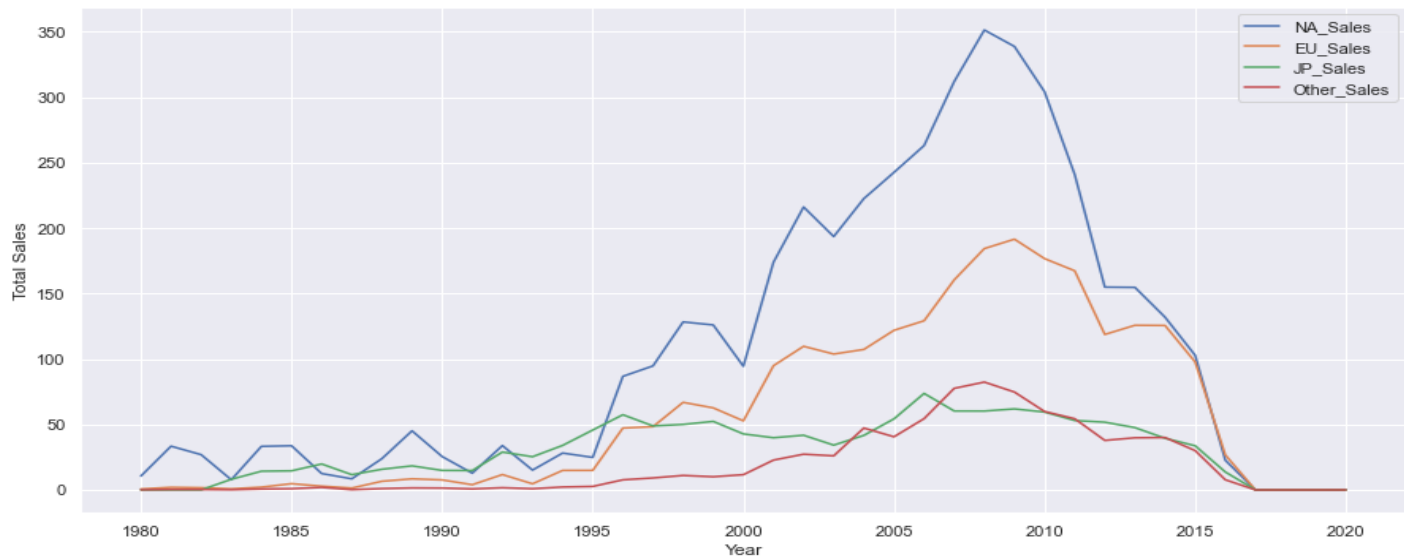


Representing the region's revenue with Pie Chart:

```
([<matplotlib.patches.Wedge at 0x1dd2b9e6be0>,
<matplotlib.patches.Wedge at 0x1dd2b7b5610>,
<matplotlib.patches.Wedge at 0x1dd2b7b5fa0>,
<matplotlib.patches.Wedge at 0x1dd2b7c5970>],
[Text(-1.0997117266962277, 0.025181703016309925, 'NA_Sales'),
Text(0.7981571088480199, -0.7569314563389279, 'EU_Sales'),
Text(0.935642530069744, 0.5784229040457237, 'JP_Sales'),
Text(0.3051488492741687, 1.0568274124882695, 'Other_Sales')],
[Text(-0.5998427600161241, 0.013735474372532685, '49.3%'),
Text(0.4353584230080108, -0.412871703457597, '27.3%'),
Text(0.510350470947133, 0.3155034022067583, '14.5%'),
Text(0.16644482687681925, 0.5764513159026924, '8.9%')])
```



Representing the region's revenue using a sales Line Plot:



- As we can see in the above graphs and charts, North America have the highest sales overall.

Reference:

1. Kaggle- [https://www.kaggle.com/datasets/gregorut/video game sales](https://www.kaggle.com/datasets/gregorut/video-game-sales)
2. Google- www.google.com
3. Up-grad EDA course-
<https://learn.upgrad.com/course/2917?courseId=25661>